

Satisfiability of the Two-Variable Fragment of First-Order Logic over Trees

Witold Charatonik^{1,*}, Emanuel Kieroński^{1,**}, and Filip Mazowiecki^{2,*}

¹ University of Wrocław

² University of Warsaw

Abstract. We consider the satisfiability problem for the two-variable fragment of first-order logic over finite unranked trees. We work with signatures consisting of some unary predicates and the binary navigational predicates \downarrow (child), \rightarrow (right sibling), and their respective transitive closures \downarrow_+ , \rightarrow_+ . We prove that the satisfiability problem for the logic containing all these predicates, $\text{FO}^2[\downarrow, \downarrow_+, \rightarrow, \rightarrow_+]$, is EXPSPACE -complete. Further, we consider the restriction of the class of structures to *singular trees*, i.e., we assume that at every node precisely one unary predicate holds. We observe that $\text{FO}^2[\downarrow, \downarrow_+, \rightarrow, \rightarrow_+]$ and even $\text{FO}^2[\downarrow_+, \downarrow]$ remain EXPSPACE -complete over finite singular trees, but the complexity decreases for some weaker logics. Namely, the logic with one binary predicate, \downarrow_+ , denoted $\text{FO}^2[\downarrow_+]$, is NEXPTIME -complete, and its guarded version, $\text{GF}^2[\downarrow_+]$, is PSPACE -complete over finite singular trees, even though both these logics are EXPSPACE -complete over arbitrary finite trees.

Keywords: two-variable logic, finite trees, satisfiability, XML

1 Introduction

Classical results from the 1930s by Church and Turing show that the satisfiability problem for first-order logic is undecidable. Moreover, undecidability can be proved even for the fragment with only three variables, FO^3 , [12]. This fact attracted the attention of researchers to the two-variable fragment, FO^2 , which turns out to be decidable [19] and NEXPTIME -complete [8]. In particular, FO^2 gained a lot of interest from computer scientists, because of its close connections to formalisms such as modal, temporal, description logics, and XML, widely used in various areas of computer science, including hardware and software verification, knowledge representation, databases, and artificial intelligence.

The expressive power of FO^2 is limited and is not sufficient to axiomatise some natural simple classes of structures, such as trees or words. It is also not possible to say, e.g., that a binary relation is transitive, an equivalence or a linear order. Thus, FO^2 over various classes of structures, in which certain relational symbols have to be interpreted in a special way, e.g., as equivalences, has been extensively studied (see, e.g., [9,10,20,14,15,16] for some results in this area).

* Supported by Polish NCN grant number DEC-2011/03/B/ST6/00346.

** Supported by Polish Ministry of Science and Higher Education grant N N206 371339.

FO^2 over words is investigated in [7]. The authors work there with signatures consisting of some unary predicates and two built-in binary predicates: *succ* for the successor relation and $<$ for its transitive closure. The resulting logic, $\text{FO}^2[\textit{succ}, <]$, is shown to have NEXPTIME-complete satisfiability problem, both over ω -words and over finite words. Actually, the lower bound can be shown for *monadic* FO^2 , i.e., without using the binary relations *succ* and $<$. The elementary complexity of FO^2 over words sharply contrasts with the non-elementary complexity of FO^3 over words which follows from [22].

In this paper we consider FO^2 over unranked trees (ordered or unordered), assuming that, beside unary symbols, signatures may include the child relation \downarrow , the right sibling relation \rightarrow , and their respective transitive closures \downarrow_+ and \rightarrow^+ . Decidability of the satisfiability problem for FO^2 over various classes of infinite trees is implied by the celebrated result by Rabin [21], that the monadic second-order theory of the binary tree is decidable. Over finite trees decidability follows from [11]. However, regarding complexity, the above mentioned results give only non-elementary upper bounds. A better upper complexity bound for the richest of the logics we consider, $\text{FO}^2[\downarrow, \downarrow_+, \rightarrow, \rightarrow^+]$, can be obtained by exploring its correspondence to XPath. In [18] it is argued that $\text{FO}^2[\downarrow, \downarrow_+, \rightarrow, \rightarrow^+]$ is expressively equivalent to a variant of Core XPath which is shown in [17] to be EXPTIME-complete. As the translation to XPath involves an exponential blowup in the size of formulas, we get this way 2-EXPTIME upper bound. Our first contribution is establishing the precise complexity of the satisfiability problem for $\text{FO}^2[\downarrow, \downarrow_+, \rightarrow, \rightarrow^+]$ over finite trees by showing that it is EXPSPACE-complete.

Worth mentioning here is the work from [4], where two-variable logics over unranked, ordered trees with additional equivalence relation on nodes, denoted \sim , is proposed. The purpose of \sim is to model XML *data values*. It is argued that this extension of $\text{FO}^2[\downarrow, \downarrow_+, \rightarrow, \rightarrow^+]$ is very hard and its decidability is left as an open problem. On the positive side, decidability of $\text{FO}^2[\downarrow, \rightarrow, \sim]$ is shown.

In the context of XML reasoning it is natural to consider also the additional semantic restriction that at a node of a tree precisely one unary predicate holds. We call trees meeting this assumption *singular trees*. In [24] an analogous restriction for finite words is considered.³ It appears that $\text{FO}^2[\textit{succ}, <]$ over finite singular words remains NEXPTIME-complete, but $\text{FO}^2[<]$ becomes NPTIME-complete. In this paper we observe a similar effect in the case of unordered trees: over singular trees, $\text{FO}^2[\downarrow_+, \downarrow]$ remains EXPSPACE-hard, and the complexity of $\text{FO}^2[\downarrow_+]$ decreases. This time the complexity drop is slightly less spectacular, as the problem is NEXPTIME-complete. We observe, however, that for NEXPTIME-hardness the ability of speaking about pairs of elements x, y in free position, i.e., such that y is neither an ascendant or descendant of x , is needed. This is not typical of logics used in computer science, as their atomic constructions usually allow to refer only to pairs of elements that lie on the same path. To capture the former kind of scenario we consider the restriction of FO^2

³ In that paper a slightly different terminology is used: the term *word* denotes a structure meeting the singularity assumption, and the term *power words* is reserved for structures that allow for multiple unary predicates holding at a single position.

to the two-variable guarded fragment, GF^2 , in which all quantifiers have to be relativised by binary predicates. We observe that the satisfiability problem for $\text{GF}^2[\downarrow_+]$ over finite singular trees is PSPACE-complete. To complete the picture we show that augmenting $\text{GF}^2[\downarrow_+]$ with any of the remaining navigational predicates leads to EXPSpace-hardness over singular trees. Thus, we establish the complexity over finite trees and over finite singular trees of all logics $\text{GF}^2[\tau_{bin}]$ and $\text{FO}^2[\tau_{bin}]$, for $\{\downarrow_+\} \subseteq \tau_{bin} \subseteq \{\downarrow, \downarrow_+, \rightarrow, \rightarrow^+\}$.

2 Preliminaries

Trees and logics. We work with signatures of the form $\tau = \tau_0 \cup \tau_{bin}$, where τ_0 is a set of unary symbols and $\tau_{bin} \subseteq \{\downarrow, \downarrow_+, \rightarrow, \rightarrow^+\}$. Over such signatures we consider two fragments of first-order logic: FO^2 , i.e., the restriction of first-order logic in which only variables x and y are available, and GF^2 being the intersection of FO^2 and the *guarded fragment*, GF [1]. GF is defined as the least set of formulas such that: (i) every atomic formula belongs to GF; (ii) GF is closed under logical connectives $\neg, \vee, \wedge, \Rightarrow$; and (iii) quantifiers are appropriately relativised by atoms, i.e., if $\varphi(\mathbf{x}, \mathbf{y})$ is a formula of GF and $\alpha(\mathbf{x}, \mathbf{y})$ is an atomic formula containing all the free variables of φ , then the formulas $\forall \mathbf{y}(\alpha(\mathbf{x}, \mathbf{y}) \Rightarrow \varphi(\mathbf{x}, \mathbf{y}))$ and $\exists \mathbf{y}(\alpha(\mathbf{x}, \mathbf{y}) \wedge \varphi(\mathbf{x}, \mathbf{y}))$ belong to GF. Atom $\alpha(\mathbf{x}, \mathbf{y})$ is called a *guard*. Equalities $x=x$ or $x=y$ are also allowed as guards.

For a given formula φ we denote by $\tau_0(\varphi)$ the set of unary symbols that appear in φ . We write $\text{FO}^2[\tau_{bin}]$ or $\text{GF}^2[\tau_{bin}]$ to denote that the only binary symbols that are allowed in signatures are those from τ_{bin} . We are interested in finite unranked tree structures, in which the interpretation of symbols from τ_{bin} is fixed: if available in the signature, \downarrow is interpreted as the child relation, \rightarrow as the right sibling relation, and \downarrow_+ and \rightarrow^+ as their respective transitive closures. If at least one of $\rightarrow, \rightarrow^+$ is interpreted in a tree then we say that this tree is *ordered*; in the opposite case we say that the tree is *unordered*.

We use $x \not\sim y$ to abbreviate the formula stating that x and y are in *free position*, i.e., that they are related by none of the binary predicates available in the signature. E.g., if we consider ordered trees over $\tau_{bin} = \{\downarrow, \downarrow_+, \rightarrow, \rightarrow^+\}$ then $x \not\sim y$ can be defined as $x \neq y \wedge \neg(x \downarrow_+ y) \wedge \neg(y \downarrow_+ x) \wedge \neg(x \rightarrow^+ y) \wedge \neg(y \rightarrow^+ x)$; for unordered trees over $\tau_{bin} = \{\downarrow_+\}$ it is just $x \neq y \wedge \neg(x \downarrow_+ y) \wedge \neg(y \downarrow_+ x)$.

Let us call the formulas specifying the relative position of a pair of elements in a tree with respect to binary predicates *order formulas*. There are ten possible order formulas: $x \downarrow y, y \downarrow x, x \downarrow_+ y \wedge \neg(x \downarrow y), y \downarrow_+ x \wedge \neg(y \downarrow x), x \rightarrow y, y \rightarrow x, x \rightarrow^+ y \wedge \neg(x \rightarrow y), y \rightarrow^+ x \wedge \neg(y \rightarrow x), x \not\sim y, x=y$. They are denoted, respectively, as: $\theta_\downarrow, \theta_\uparrow, \theta_{\downarrow\downarrow+}, \theta_{\uparrow\uparrow+}, \theta_\rightarrow, \theta_\leftarrow, \theta_{\Rightarrow+}, \theta_{\Leftarrow+}, \theta_\neq, \theta_=$. Let Θ be the set of these ten formulas.

A structure over a signature $\tau = \tau_0 \cup \tau_{bin}$ is *singular* if at every element of this structure precisely one unary predicate from τ_0 holds. We say that a formula φ is *singularly satisfiable* (over a class of structures \mathcal{C}) if there exists a singular model of φ (from \mathcal{C}).

We use symbol \mathfrak{T} (possibly with sub- or superscripts) to denote tree structures. For a given tree \mathfrak{T} we denote by T its universe. A *tree frame* is a tree

over a signature containing no unary predicates. We say that a formula φ is (singularly) satisfiable over a tree frame \mathcal{T} if $\mathfrak{T} \models \varphi$ for some (singular) \mathfrak{T} such that \mathcal{T} is the restriction of \mathfrak{T} to binary symbols.

Normal form. We say that an $\text{FO}^2[\downarrow, \downarrow_+, \rightarrow, \rightarrow^+]$ formula φ is in *normal form* if $\varphi = \forall xy \chi(x, y) \wedge \bigwedge_{i \in I} \forall x (\lambda_i(x) \Rightarrow \exists y (\eta_i(x, y) \wedge \psi_i(x, y)))$, for some index set I , where $\chi(x, y)$ is quantifier-free, $\lambda_i(x)$ is an atomic formula $a(x)$ for some unary symbol a , $\psi_i(x, y)$ is a boolean combination of unary atomic formulas, and $\eta_i(x, y)$ is an order formula. Please note, that in χ the equality symbol may be used, e.g., we can enforce that a model contains at most one node satisfying a : $\forall xy (a(x) \wedge a(y) \Rightarrow x=y)$. The following lemma can be proved in a standard fashion (cf. e.g., [16]).

Lemma 1. *Let φ be an $\text{FO}^2[\downarrow, \downarrow_+, \rightarrow, \rightarrow^+]$ formula over a signature τ and let \mathcal{T} be a tree frame. There exists a polynomially computable $\text{FO}^2[\downarrow, \downarrow_+, \rightarrow, \rightarrow^+]$ normal form formula φ' over signature τ' consisting of τ and some additional unary symbols, such that φ is satisfiable over \mathcal{T} (singularly satisfiable over \mathcal{T}) iff φ' is satisfiable over \mathcal{T} (satisfiable over \mathcal{T} in a model that restricted to τ is singular).*

Consider a conjunct $\varphi_i = \forall x (\lambda_i(x) \Rightarrow \exists y (\eta_i(x, y) \wedge \psi_i(x, y)))$ of a normal form $\text{FO}^2[\downarrow, \downarrow_+, \rightarrow, \rightarrow^+]$ formula φ . Let $\mathfrak{T} \models \varphi$, and let $v \in T$ be an element such that $\mathfrak{T} \models \lambda_i[v]$. Then an element $w \in T$ such that $\mathfrak{T} \models \eta_i[v, w] \wedge \psi_i[v, w]$ is called a *witness* for v and φ_i . Sometimes, b is called an *upper* witness if $\eta_i(x, y) \models y \downarrow_+ x$, a *lower* witness if $\eta_i(x, y) \models x \downarrow_+ y$, and a *free* witness if $\eta_i(x, y) \models x \not\sim y$.

Types. A (atomic) *1-type*, over a signature $\tau = \tau_0 \cup \tau_{bin}$, is a subset of τ_0 . We often identify a 1-type α with the formula $\bigwedge_{a \in \alpha} a(x) \wedge \bigwedge_{a \notin \alpha} \neg a(x)$. For a given τ -tree \mathfrak{T} , and $v \in T$, we denote by $\text{tp}^{\mathfrak{T}}(v)$ the 1-type *realized* by v , i.e., the unique 1-type α such that $\mathfrak{T} \models \alpha[v]$.

A *full type* is a function $\bar{\alpha} : \Theta \rightarrow \mathcal{P}(\tau_0)$, such that $\bar{\alpha}(\theta_{\uparrow})$, $\bar{\alpha}(\theta_{\rightarrow})$, $\bar{\alpha}(\theta_{\leftarrow})$ are singletons or empty, $\bar{\alpha}(\theta_{=})$ is a singleton, and if $\bar{\alpha}(\theta_{\uparrow})$ (respectively $\bar{\alpha}(\theta_{\downarrow})$, $\bar{\alpha}(\theta_{\leftarrow})$, $\bar{\alpha}(\theta_{\rightarrow})$) is empty then $\bar{\alpha}(\theta_{\uparrow\uparrow+})$ (respectively $\bar{\alpha}(\theta_{\downarrow\downarrow+})$, $\bar{\alpha}(\theta_{\leftarrow+})$, $\bar{\alpha}(\theta_{\rightarrow+})$) is also empty. We employ the following convention: for a given full type $\bar{\alpha}$ we denote by α the unique member of $\bar{\alpha}(\theta_{=})$. For a given τ -tree \mathfrak{T} , and $v \in T$, we denote by $\text{ftp}^{\mathfrak{T}}(v)$ the full type *realized* by v , i.e., the unique full type $\bar{\alpha}$, such that α is the 1-type of v , and for all $\theta \in \Theta$ we have that $\bar{\alpha}(\theta) = \{\text{tp}^{\mathfrak{T}}(w) : \mathfrak{T} \models \theta[v, w]\}$.

A *reduced full type* is a tuple (α, A, B, F) , where α is a 1-type and A, B, F are sets of 1-types. Reduced full types are used to keep information recorded in full types in a slightly (lossy) compressed form. Let $\text{ftp}^{\mathfrak{T}}(v) = \bar{\alpha}$. By $\text{rftp}^{\mathfrak{T}}(v)$ we denote the reduced full type *realized* by v , i.e., the reduced full type (α, A, B, F) , such that $A = \bar{\alpha}(\theta_{\uparrow}) \cup \bar{\alpha}(\theta_{\uparrow\uparrow+})$, $B = \bar{\alpha}(\theta_{\downarrow}) \cup \bar{\alpha}(\theta_{\downarrow\downarrow+})$ and $F = \bar{\alpha}(\theta_{\rightarrow}) \cup \bar{\alpha}(\theta_{\leftarrow}) \cup \bar{\alpha}(\theta_{\rightarrow+}) \cup \bar{\alpha}(\theta_{\leftarrow+}) \cup \bar{\alpha}(\theta_{\neq})$. Note that α denotes the 1-type of v , and, informally speaking, A is the set of 1-types of elements realized *above* v , B is the set of 1-types of elements realized *below* v , and F is the set of 1-types of the siblings of v and the elements realized in *free* position to v .

Note that the number of 1-types is bounded exponentially, and the numbers of full types and reduced full types are bounded doubly exponentially in the size of the signature.

For a given normal form $\text{FO}^2[\downarrow, \downarrow_+, \rightarrow, \rightarrow^+]$ formula φ and a full type $\bar{\alpha}$, we say that $\bar{\alpha}$ is φ -consistent if an element realizing $\bar{\alpha}$ cannot be a member of a pair violating the universal conjunct $\forall xy\chi(x, y)$ of φ , and has all witnesses required by φ . Formally, $\bar{\alpha}$ is φ -consistent if for every $\theta \in \Theta$, and every $\alpha' \in \bar{\alpha}(\theta)$ we have $\alpha(x) \wedge \alpha'(y) \wedge \theta(x, y) \models \chi(x, y) \wedge \chi(y, x)$, and for every conjunct $\forall x(\lambda_i(x) \Rightarrow \exists y(\eta_i(x, y) \wedge \psi_i(x, y)))$ of φ , such that $\alpha(x) \models \lambda_i(x)$, there exists a 1-type $\alpha' \in \bar{\alpha}(\eta_i)$ such that $\alpha(x), \alpha'(y) \models \psi_i(x, y)$. A proof of the following proposition is straightforward.

Proposition 1. *Let \mathfrak{T} be a tree and let φ be a normal form $\text{FO}^2[\downarrow, \downarrow_+, \rightarrow, \rightarrow^+]$ -formula. Then $\mathfrak{T} \models \varphi$ iff every full type realized in \mathfrak{T} is φ -consistent.*

We say that a full type $\bar{\alpha}$ is *combined* of two full types $\bar{\alpha}_1$ and $\bar{\alpha}_2$ if $\alpha = \alpha_1 = \alpha_2$ and for each $\theta \in \Theta$ we have $\bar{\alpha}(\theta) = \bar{\alpha}_1(\theta)$ or $\bar{\alpha}(\theta) = \bar{\alpha}_2(\theta)$. Also the following fact is immediate.

Proposition 2. *Let φ be a normal form $\text{FO}^2[\downarrow, \downarrow_+, \rightarrow, \rightarrow^+]$ -formula, and let $\bar{\alpha}$ be a full type combined of two φ -consistent full types $\bar{\alpha}_1, \bar{\alpha}_2$. Then $\bar{\alpha}$ is φ -consistent.*

3 Finite ordered trees

This section is devoted to a proof of the following theorem.

Theorem 1. *The satisfiability problem for $\text{FO}^2[\downarrow, \downarrow_+, \rightarrow, \rightarrow^+]$ over finite trees is EXSPACE-complete.*

The crucial fact is that every satisfiable formula has a model of exponentially bounded depth and degree. We prove this in two steps, and present a procedure looking for such small models, working in alternating exponential time.

Short paths. First, let us see how the paths of a model can be shortened.

Lemma 2. *Let φ be a normal form $\text{FO}^2[\downarrow, \downarrow_+, \rightarrow, \rightarrow^+]$ formula, \mathfrak{T} its model, and $v, w \in T$ two nodes of \mathfrak{T} , such that $\mathfrak{T} \models v \downarrow_+ w$ and $\text{rftp}^{\mathfrak{T}}(v) = \text{rftp}^{\mathfrak{T}}(w)$. Then the tree \mathfrak{T}' , obtained from \mathfrak{T} by replacing the subtree rooted at v by the subtree rooted at w , is a model of φ .*

Proof. It can be verified that for every $u \in T'$, if $u \neq w$ then $\text{ftp}^{\mathfrak{T}'}(u) = \text{ftp}^{\mathfrak{T}}(u)$, and that $\text{ftp}^{\mathfrak{T}'}(w)$ is combined of $\text{ftp}^{\mathfrak{T}}(v)$ and $\text{ftp}^{\mathfrak{T}}(w)$. Thus, by Propositions 1 and 2, all types realized in \mathfrak{T}' are φ -consistent, and $\mathfrak{T}' \models \varphi$ by Proposition 1. \square

Using the above lemma we can successively shorten \downarrow -paths in a model of a normal form formula φ obtaining after a finite number of steps a model of φ in which on every path only distinct reduced full types are realized. Even though there are potentially doubly exponentially many reduced full types it can be shown that such a model has exponentially bounded \downarrow_+ -paths.

Lemma 3. *Let φ be a normal form $\text{FO}^2[\downarrow, \downarrow_+, \rightarrow, \rightarrow^+]$ formula satisfied in a finite tree. Then there exists a tree model of φ whose every \downarrow -path has length bounded by $3 \cdot (2^{2 \cdot |\tau_0(\varphi)|})$, exponentially in $|\varphi|$.*

Proof. Let $\mathfrak{T} \models \varphi$ be a tree in which on every \downarrow -path only distinct full types are realized and let v_1, v_2, \dots, v_k be a \downarrow -path in \mathfrak{T} . Observe that the sets A, B, F in reduced full types of v_i behave monotonically. More precisely, if $(\alpha_i, A_i, B_i, F_i)$ is the reduced full type realized by v_i , for $1 \leq i \leq k$, then for $i < j$ we have $A_i \subseteq A_j$, $B_i \supseteq B_j$ and $F_i \subseteq F_j$. Thus along the path each of the sets A, B, F is modified at most $2^{|\tau_0(\varphi)|}$ times (since this is the number of possible 1-types). The number of reduced full types with fixed A, B, F is equal to the number of 1-types, so the length of each path is bounded as required. \square

Small degree. Now we observe that to provide all witnesses for $\forall\exists$ conjuncts of φ we only need nodes with at most exponential degree.

Lemma 4. *Let φ be a normal form $\text{FO}^2[\downarrow, \downarrow_+, \rightarrow, \rightarrow^+]$ formula and let $\mathfrak{T} \models \varphi$. Then there exists a model $\mathfrak{T}' \models \varphi$ in which the number of successors of each node is bounded by $4 \cdot 2^{2 \cdot |\tau_0(\varphi)|}$. Moreover \mathfrak{T}' can be obtained by removing from \mathfrak{T} some number of elements (together with the subtrees rooted at them).*

Proof. We show first how to decrease the degree of a single node of \mathfrak{T} . Let v be a node of \mathfrak{T} of full type $\bar{\alpha}_v$, and let U be the set of the children of v . For every element $u \in U$ let $\bar{\alpha}_u$ be its full type. We are going to mark some important elements of U and then remove all subtrees rooted at unmarked ones producing a model $\mathfrak{T}''' \models \varphi$. First, for every 1-type α , if α is realized in U precisely once then mark this realisation; if α is realized more than once then mark the minimal and the maximal (with respect to \rightarrow^+) realisations of α . Further, for every 1-type α , let $U_\alpha = \{u \in U \mid \alpha \in \bar{\alpha}_u(\theta_\downarrow) \cup \bar{\alpha}_u(\theta_{\downarrow+})\}$. For each α mark $\min(2, |U_\alpha|)$ elements of U_α . Note that so far we have marked at most $4 \cdot |2^{\tau_0(\varphi)}|$ elements of U . Assume that these (listed according to \rightarrow^+) are: u_1, \dots, u_k . We call them *primarily marked* elements, and denote their set by U_P .

Consider the tree \mathfrak{T}'' obtained from \mathfrak{T} by removing the subtrees rooted at elements of $U \setminus U_P$. It can be verified that elements from $T'' \setminus U_P$ retain in \mathfrak{T}'' their full types from \mathfrak{T} . Unfortunately, the \rightarrow -connections among the elements of U_P in \mathfrak{T}'' may be inconsistent with φ . To fix this problem we mark some additional elements of U (at most exponentially many) between u_i and u_{i+1} , for all i .⁴ For every i , consider the \rightarrow -chain C of elements of \mathfrak{T} between u_i and u_{i+1} . If C is empty then u_{i+1} is \rightarrow -successor of u_i and there is nothing to do. Otherwise, let α be the 1-type of the successor w of u_i . Find the maximal (with respect to \rightarrow^+) element w' of type α in C , and mark it. The elements between u_i and w' will never be marked, so w' will become the \rightarrow -successor of u_i in the final model \mathfrak{T}''' . Thus, u_i will retain in its full type its $\bar{\alpha}_{u_i}(\theta_{\rightarrow})$ (singleton) set, and, due to our strategy of primarily marking maximal realisations of 1-types, also its

⁴ Actually, this fragment of the construction combined with some earlier parts, reproduces the small model theorem for FO^2 over words.

$\bar{\alpha}_{u_i}(\theta_{\rightarrow+})$ set. This is not necessarily true for w' and its (singleton) $\bar{\alpha}(\theta_{\leftarrow})$ set, and $\bar{\alpha}_{w'}(\theta_{\leftarrow+})$ set. However, these sets will be equal, respectively, to $\bar{\alpha}_w(\theta_{\leftarrow})$, and $\bar{\alpha}_w(\theta_{\leftarrow+})$ sets of w , which means that the full type of w' in \mathfrak{T}''' will be combined of two full types (of w and w') from \mathfrak{T} . We proceed recursively with the \rightarrow -chain of elements between w' and u_{i+1} .

Note that the number of elements between u_i and u_{i+1} which are marked during this process is bounded by the number of 1-types. Thus we mark in total at most $4 \cdot 2^{|\tau_0(\varphi)|} \cdot 2^{|\tau_0(\varphi)|}$ elements of U , as required in the statement of this lemma. Let us denote the set of the marked elements U_M . We construct \mathfrak{T}''' by removing from \mathfrak{T} all subtrees rooted at elements of $U \setminus U_M$. It can be verified that all elements from $T''' \setminus U_M$ retain their full types from \mathfrak{T} , and that the full types of elements from U_M in \mathfrak{T}''' are either retained from \mathfrak{T} or are combined of pairs of full types in \mathfrak{T} of elements from U . By Proposition 1 we have that $\mathfrak{T}''' \models \varphi$.

The desired model \mathfrak{T}' can be obtained by applying the described procedure in depth-first manner. \square

Alternating procedure and complexity. We are ready to design a procedure checking if a given $\text{FO}^2[\downarrow, \downarrow_+, \rightarrow, \rightarrow^+]$ formula φ has a finite tree model. By Lemma 1 we may assume that φ is in normal form. By Lemma 3 and Lemma 4 we may restrict our attention to models in which the length of each path and the degree of each node are bounded exponentially in $|\varphi|$. We present an alternating procedure working in exponential time. This justifies that the problem is in EXPSPACE since, by [5], $\text{EXPSPACE} = \text{AEXPTIME}$. The procedure first guesses the full type of the root and then guesses the full types of its children, checking if the information recorded in the full types is locally consistent, and if each full type is φ -consistent. Further, it works in a loop, universally choosing one of the types of the children and proceeding similarly.

Procedure $\text{FO}^2[\downarrow, \downarrow_+, \rightarrow, \rightarrow^+]$ -sat-test

input: an $\text{FO}^2[\downarrow, \downarrow_+, \rightarrow, \rightarrow^+]$ normal form formula φ

- let $\text{maxdepth} := 3 \cdot |2^{\tau_0(\varphi)}|^2$; let $\text{maxdegree} := 4 \cdot 2^{2 \cdot |\tau_0(\varphi)|}$;
- let $\text{level} := 0$;
- **guess** a full type $\bar{\alpha}$ such that $\bar{\alpha}(\theta_{\uparrow}) = \bar{\alpha}(\theta_{\uparrow+}) = \bar{\alpha}(\theta_{\rightarrow}) = \bar{\alpha}(\theta_{\rightarrow+}) = \bar{\alpha}(\theta_{\leftarrow}) = \bar{\alpha}(\theta_{\leftarrow+}) = \bar{\alpha}(\theta_{\nearrow}) = \emptyset$;
- **while** $\text{level} < \text{maxdepth}$ **do**
 - if $\bar{\alpha}$ is not φ -consistent then **reject**
 - if $\bar{\alpha}(\theta_{\downarrow}) \cup \bar{\alpha}(\theta_{\downarrow+}) = \emptyset$ then **accept**
 - **guess** an integer $1 \leq k \leq \text{maxdegree}$;
 - for $1 \leq i \leq k$ **guess** a full type type $\bar{\alpha}_i$;
 - if not *locally-consistent*($\bar{\alpha}, \bar{\alpha}_1, \dots, \bar{\alpha}_k$) then **reject**;
 - $\text{level} := \text{level} + 1$;
 - **universally choose** $1 \leq i \leq k$; let $\bar{\alpha} = \bar{\alpha}_i$;
- **endwhile**
- **reject**

The function *locally-consistent* checks whether, from a local point of view, a tree may have a node of full type $\bar{\alpha}$ whose children, listed from left to right, have full types $\bar{\alpha}_1, \dots, \bar{\alpha}_k$. Namely, it returns **true** if and only if all of the following conditions hold:

Horizontal conditions:

- (h1) $\bar{\alpha}_i(\theta_{\leftarrow}) = \{\alpha_{i-1}\}$ for $i > 1$; $\bar{\alpha}_1(\theta_{\leftarrow}) = \emptyset$;
- (h2) $\bar{\alpha}_i(\theta_{\rightarrow}) = \{\alpha_{i+1}\}$ for $i < k$; $\bar{\alpha}_k(\theta_{\rightarrow}) = \emptyset$;
- (h3) $\bar{\alpha}_i(\theta_{\leftarrow+}) = \bar{\alpha}_{i-1}(\theta_{\leftarrow}) \cup \bar{\alpha}_{i-1}(\theta_{\leftarrow+})$ for $i > 1$; $\bar{\alpha}_1(\theta_{\leftarrow+}) = \emptyset$;
- (h4) $\bar{\alpha}_i(\theta_{\rightarrow+}) = \bar{\alpha}_{i+1}(\theta_{\rightarrow}) \cup \bar{\alpha}_{i+1}(\theta_{\rightarrow+})$ for $i < k$; $\bar{\alpha}_k(\theta_{\rightarrow+}) = \emptyset$;

Vertical conditions:

- (v1) $\bar{\alpha}(\theta_{\downarrow}) = \{\alpha_1, \dots, \alpha_k\}$;
- (v2) $\bar{\alpha}_i(\theta_{\uparrow}) = \{\alpha\}$ for $1 \leq i \leq k$;
- (v3) $\bar{\alpha}(\theta_{\downarrow+}) = \bigcup_{1 \leq i \leq k} (\bar{\alpha}_i(\theta_{\downarrow}) \cup \bar{\alpha}_i(\theta_{\downarrow+}))$;
- (v4) $\bar{\alpha}_i(\theta_{\uparrow+}) = \bar{\alpha}(\theta_{\uparrow}) \cup \bar{\alpha}(\theta_{\uparrow+})$ for $1 \leq i \leq k$;

Free conditions:

- (f1) $\bar{\alpha}_i(\theta_{\nearrow}) = \bigcup_{j \neq i} (\bar{\alpha}_j(\theta_{\downarrow}) \cup \bar{\alpha}_j(\theta_{\downarrow+})) \cup \bar{\alpha}(\theta_{\leftarrow+}) \cup \bar{\alpha}(\theta_{\leftarrow}) \cup \bar{\alpha}(\theta_{\rightarrow}) \cup \bar{\alpha}(\theta_{\rightarrow+}) \cup \bar{\alpha}(\theta_{\nearrow})$ for $1 \leq i \leq k$.

Lemma 5. *Procedure $\text{FO}^2[\downarrow, \downarrow_+, \rightarrow, \rightarrow^+]\text{-sat-test}$ accepts its input φ if and only if φ is satisfied in a finite tree.*

A matching EXPSpace-lower bound follows from [13], where it was shown that a restricted variant of the two-variable guarded fragment with some unary predicates and a single binary predicate that is interpreted as a transitive relation is EXPSpace-hard. It is not hard to see that the proof presented there works fine (actually, it is even more natural) if we restrict the class of admissible structures to (finite) trees. Thus we get the following corollary.

Corollary 1. *Over finite trees the satisfiability problem for each logic between $\text{GF}^2[\downarrow_+]$ and $\text{FO}^2[\downarrow, \downarrow_+, \rightarrow, \rightarrow^+]$ is EXPSpace-complete.*

4 Singular finite trees

We start this section with establishing the complexity of $\text{FO}^2[\downarrow_+]$.

Theorem 2. *The satisfiability problem for $\text{FO}^2[\downarrow_+]$ over finite singular trees is NEXPTIME-complete.*

To show the upper bound we observe that every singularly satisfiable formula has a singular model whose all paths are bounded polynomially. This fact is a generalisation of Theorem 2.1.1 from [24], that every $\text{FO}^2[<]$ formula φ , singularly satisfiable over finite words, has a finite singular model with polynomially many elements. Actually, our work is strongly influenced by the construction from [24], and, generally, can be seen as its adaptation to the case of trees. We

describe here all the required constructions, but omit some proofs, as many of them are obtained by obvious adjustments of the corresponding proofs for the case of words. Thus, in order to fully understand all the details, we advise the reader to familiarise with Chapter 2 of [24].

Before going further we discuss the main differences with the case of words. The main idea from [24] is to show that for a given singular word $\mathfrak{W} \models \varphi$, a letter $a \in \tau_0$ and a given subformula $\xi(x)$ of φ there exists a division of \mathfrak{W} into polynomially many segments in which, at elements satisfying a , the value of $\xi(x)$ is constant. In our case the role of those segments is played by *slices*, i.e., connected components of trees. We show that each path intersects polynomially many slices. In [24] *left* and *right* witnesses are considered. In our case they correspond to *upper* and *lower* witnesses (which, however, in contrast to the case of words, are not necessarily linearly ordered), but we must also deal with *free* witnesses. Finally, the small model is constructed by picking at most three witnesses for each slice. As the total number of considered slices in a tree may be exponential we have to be careful at this point, to avoid choosing too many witnesses from a single path.

Now we turn to technical details. Recall that in the current scenario we have four order formulas $x \downarrow_+ y$, $x=y$, $y \downarrow_+ x$ and $x \not\downarrow_+ y$. We also use a shortcut: $x \downarrow_* y = x \downarrow_+ y \vee x=y$. The normal form from Lemma 1 is not very useful since it introduces fresh unary predicates that destroy singularity of models. Thus, we only slightly adjust formulas by converting them to existential negation form (ENNF). A formula $\varphi \in \text{FO}^2[\downarrow_+]$ is in ENNF if it does not contain any universal quantifier, and negations only appear in front of unary predicates or existential quantifiers. Negations in front of order formulas are not allowed. Obviously, any formula $\varphi \in \text{FO}^2[\downarrow_+]$ is equivalent to a formula in ENNF of size at most $2|\varphi|$.

We may view our formulas as positive boolean combinations of order formulas and formulas with at most one free variable.

Proposition 3. *Let $\varphi \in \text{FO}^2[\downarrow_+]$ be a formula in ENNF. Then there exists a number $s \in \mathbb{N}$, a positive boolean formula β in variables $Z_{\downarrow_+}, Z_-, Z_{\uparrow_+}, Z_{\not\downarrow_+}, X_1, \dots, X_s$, and formulas $\varphi_1, \dots, \varphi_s \in \text{FO}^2[\downarrow_+]$ in ENNF, each with at most one free variable, such that $\varphi = \beta(x \downarrow_+ y, x=y, y \downarrow_+ x, x \not\downarrow_+ y, \varphi_1, \dots, \varphi_s)$. Moreover $\varphi \equiv (x \downarrow_+ y \wedge \varphi_{\downarrow_+}) \vee (x=y \wedge \varphi_{=}) \vee (y \downarrow_+ x \wedge \varphi_{\uparrow_+}) \vee (x \not\downarrow_+ y \wedge \varphi_{\not\downarrow_+})$ where $\varphi_{\downarrow_+} = \beta(\top, \perp, \perp, \perp, \varphi_1, \dots, \varphi_s)$, and φ_{θ} is analogously defined for the remaining θ -s.*

For a finite tree \mathfrak{T} and a set of nodes $P \subseteq T$ we define $\max(P)$ as the set of the maximal nodes from P and $\min(P)$ as the set of the minimal nodes from P , with respect to the order relation \downarrow_+ . For example $\max(T)$ is the set of the leaves and $\min(T)$ is the singleton consisting of the root of \mathfrak{T} .

Lemma 6. *Let $\zeta_1(y), \dots, \zeta_t(y)$ be $\text{FO}^2[\downarrow_+]$ formulas with y as the only free variable and in ENNF, and let \mathfrak{T} be a finite singular tree. Let β be a positive boolean formula in the variables $Z_{\downarrow_+}, Z_-, Z_{\uparrow_+}, Z_{\not\downarrow_+}, Y_1, \dots, Y_t$, let $\psi(x, y) = \beta(x \downarrow_+ y, x=y, y \downarrow_+ x, x \not\downarrow_+ y, \zeta_1(y), \dots, \zeta_t(y))$, and let $\varphi(x) = \exists y \psi(x, y)$. Let $P' := \{u \in T \mid \mathfrak{T} \models \psi_{\downarrow_+}[u, v], \text{ for some } v \text{ s.t. } u \downarrow_+ v\}$, $Q' := \{u \in T \mid \mathfrak{T} \models \psi_{\uparrow_+}[u, v], \text{ for some } v \text{ s.t. } v \downarrow_+ u\}$, $R' := \{u \in T \mid \mathfrak{T} \models \psi_{\not\downarrow_+}[u, v], \text{ for some } v \text{ s.t. } u \not\downarrow_+ v\}$. Set $P =$*

$\max(P')$ and $Q = \min(Q' \cup R')$. Then for all $u \in \mathfrak{T}$, $\mathfrak{T} \models \varphi[u]$ iff there exists $p \in P$ s.t. $u \downarrow_* p$ or there exists $q \in Q$ s.t. $q \downarrow_* u$ or $\mathfrak{T} \models \psi_{|x=y}[u, u]$.

Remark. Notice that on every path in \mathfrak{T} there is at most one point from P and at most one point from Q .

Let $a \in \tau_0$ be a letter, \mathfrak{T} a finite singular tree, and S a set of nodes of \mathfrak{T} . Then by S^a we denote the set of nodes in S where the letter a occurs. We also say that S is a tree *slice* iff it induces a connected (with respect to the symmetric closure of the child relation \downarrow) subgraph of \mathfrak{T} .

Lemma 7. *Let $\varphi \in \text{FO}^2[\downarrow_+]$ be a formula in ENNF and with one free variable, let \mathfrak{T} be a finite singular tree, and let $a \in \tau_0$. There is a set $S \subseteq T$ which is a union of tree slices in \mathfrak{T} such that: for every $u \in T^a$ we have $\mathfrak{T} \models \varphi[u]$ iff $u \in S$; and every path in \mathfrak{T} intersects at most $|\varphi|^2$ tree slices from S .*

The proof is inductive: if $\varphi = \exists y \psi(x, y)$, for $\varphi(x, y) = \beta(x \downarrow_+ y, x=y, y \downarrow_+ x, x \not\downarrow y, \xi_1(x), \dots, \xi_s(x), \zeta_1(y), \dots, \zeta_t(y))$, then we consider the slices obtained inductively for the formulas $\xi_\sigma(x)$. The slices for different σ -s may overlap. Their endpoints determine a more refined division into slices, such that in each slice, on nodes carrying a , the values of all $\xi_\sigma(x)$ are constant. In each such slice we apply Lemma 6 to introduce new divisions. Now arguments and calculations similar as in the proof of the corresponding Lemma 2.1.10 from [24] lead to the desired claim.

Lemma 8. *Let φ be an $\text{FO}^2[\downarrow_+]$ formula over a signature τ . If φ is satisfied in a singular tree, then φ is also satisfied in a singular tree, in which the length of every path is bounded by $6 \cdot |\tau| \cdot |\varphi|^3$.*

Proof. We assume that φ is in ENNF. Let $\mathfrak{T} \models \varphi$ be singular, and let $\varphi_1, \dots, \varphi_k$ be the subformulas of φ of the form $\exists x \psi$ for some variable x and some formula ψ . For every $\kappa \in [1, \dots, k]$ we use Proposition 3 to find a positive boolean formula such that $\psi_\kappa(x, y) = \beta(x \downarrow_+ y, x=y, y \downarrow_+ x, x \not\downarrow y, \xi_1(x), \dots, \xi_s(x), \zeta_1(y), \dots, \zeta_t(y))$. For every $a \in \tau_0$ and every $\sigma \in [1, \dots, s]$ let S_σ^a be a set as in Lemma 7 applied to the formula $\xi_\sigma(x)$ and a , where every path intersects at most $|\xi_\sigma|^2$ tree slices from S_σ^a . Thus there is a set \mathcal{I}_κ^a of tree slices I such that: every path in \mathfrak{T} intersects at most $2 \cdot \sum_{\sigma \in [1, s]} |\xi_\sigma|^2$ of them; $\bigcup_{I \in \mathcal{I}_\kappa^a} I = \mathfrak{T}$; and there are $\xi_1^I, \dots, \xi_s^I \in \{\top, \perp\}$ such that $\mathfrak{T} \models \xi_\sigma[u]$ iff $\xi_\sigma^I = \top$ for every $u \in I$ satisfying $a[u]$. For each $I \in \mathcal{I}_\kappa^a$ we consider the formula $\varphi_\kappa^I = \exists y \psi_\kappa^I(x, y)$, where $\psi_\kappa^I(x, y) = \beta(x \downarrow_+ y, x=y, y \downarrow_+ x, x \not\downarrow y, \xi_1^I, \dots, \xi_s^I, \zeta_1(y), \dots, \zeta_t(y))$. Let $P'' := \{v \in \mathfrak{T} \mid \mathfrak{T} \models \psi_\kappa^I[x \downarrow_+ y][u, v] \text{ for all } u \downarrow_+ v\}$, $Q'' := \{v \in \mathfrak{T} \mid \mathfrak{T} \models \psi_\kappa^I[y \downarrow_+ x][u, v] \text{ for all } v \downarrow_+ u\}$, $R'' := \{v \in \mathfrak{T} \mid \mathfrak{T} \models \psi_\kappa^I[x \not\downarrow y][u, v] \text{ for all } u \not\downarrow v\}$, and let $P_I = \max(P'')$, $Q_I = \min(Q'')$, $R_I = \max(R'')$. Let $T_\kappa^a = \bigcup_{I \in \mathcal{I}_\kappa^a} (P_I \cup Q_I \cup R_I)$, $T' = \bigcup_{\kappa \in [1, k], a \in \tau_0} T_\kappa^a$, and let \mathfrak{T}' be the restriction of \mathfrak{T} to T' . Lemma 6 can be used to prove that $\mathfrak{T}' \models \varphi$. Also it can be shown that the paths of \mathfrak{T}' are bounded as required. \square

Corollary 2. *Let φ be a singularly satisfiable $\text{FO}^2[\downarrow_+]$ formula. Then it is satisfied in a singular tree whose number of nodes is exponential in $|\varphi|$.*

Proof. Let $\mathfrak{T} \models \varphi$ be a singular tree over the signature τ , let \mathcal{T} be the frame of \mathfrak{T} . By Lemma 8 we may assume that all its paths are bounded polynomially. Let φ' be the normal form formula over signature τ' from the statement of Lemma 1. By that lemma φ' is satisfiable in a model \mathfrak{T}' based on the frame \mathcal{T} . By Lemma 4 we can remove some subtrees from \mathfrak{T}' to obtain a model $\mathfrak{T}'' \models \varphi'$ with exponentially bounded degree. Again by Lemma 1, the restriction of \mathfrak{T}'' to the original signature τ is a singular model. As its paths are bounded polynomially and the degree of nodes is bounded exponentially, the total number of nodes is bounded exponentially in $|\varphi|$ as required. \square

Corollary 2 justifies the upper bound from Theorem 2, since for a given φ we can nondeterministically guess its exponential model and then verify it.

The exponential bound on the degree of nodes in singular models of $\text{FO}^2[\downarrow_+]$ formulas is essentially optimal. Indeed, let us see that there exists a formula of size polynomial in n in whose every model the root has 2^n children. We use unary predicates $root, elem, b_0, \dots, b_{n-1}$, and say that all elements in $elem$ are children of the root: $\forall x (root(x) \Leftrightarrow \neg \exists y y \downarrow_+ x) \wedge \forall x (elem(x) \Leftrightarrow (\neg \exists y (y \downarrow_+ x \wedge \neg root(y))))$. We think that each v in $elem$ encodes a number $0 \leq N(v) < 2^n$ such that the i -th bit in its binary representation is 1 iff the formula $\delta_i(x) = \exists y (x \downarrow_+ y \wedge b_i(y))$ is satisfied at v . In a standard way we can now write a formula $first(x)$ which says that $N(x) = 0$, a formula $last(x)$ stating that $N(x) = 2^n - 1$, and a formula $succ(x, y)$ saying that $N(y) = N(x) + 1$. Now the formula $\exists x first(x) \wedge \forall x (\neg last(x) \Rightarrow \exists y succ(x, y))$ is as required. This idea can be easily employed to obtain NEXPTIME-lower bound in Theorem 2.

It appears, that the ability of speaking about pairs of nodes in free position is crucial for NEXPTIME-hardness. Indeed if we allow only guarded formulas, we get PSPACE complexity. The upper bound in the following theorem can be proved by bounding polynomially not only the length of the paths but also the degree of the nodes in models of $\text{GF}^2[\downarrow_+]$ formulas.

Theorem 3. *The satisfiability problem for $\text{GF}^2[\downarrow_+]$ over finite singular trees is PSPACE-complete.*

Finally we show that augmenting $\text{GF}^2[\downarrow_+]$ with any of the remaining binary navigational predicates leads to EXPSpace-lower bound over singular trees.

Theorem 4. *The satisfiability problem over singular trees for each of the logics $\text{GF}^2[\downarrow_+, \downarrow]$, $\text{GF}^2[\downarrow_+, \rightarrow]$, $\text{GF}^2[\downarrow_+, \rightarrow^+]$ is EXPSpace-hard.*

5 Future work

One possible direction of a further research could be investigating the case in which infinite trees are admitted as models. It seems that the complexity results we have obtained for finite trees can be transferred to this case without major difficulties. It could be interesting to examine also the cases in which τ_{bin} contains \downarrow but does not contain \downarrow_+ . A related result is obtained in [6], where NEXPTIME-completeness of FO^2 with counting quantifiers and arbitrary number of binary

symbols, of which fixed two have to be interpreted as child relations in two trees. The trees considered in [6] are, however, ranked and unordered.

Acknowledgement. Similar results were obtained independently in [3]. The two works were merged into a single paper [2].

References

1. H. Andréka, J. van Benthem, and I. Németi. Modal languages and bounded fragments of predicate logic. *Journal of Philosophical Logic*, 27:217–274, 1998.
2. Saguy Benaim, Michael Benedikt, Witold Charatonik, Emanuel Kieroński, Rastislav Lenhardt, Filip Mazowiecki, and James Worrell. Complexity of two-variable logic on finite trees. *Accepted for ICALP*, 2013.
3. Saguy Benaim, Michael Benedikt, Rastislav Lenhardt, and James Worrell. Controlling the depth, size, and number of subtrees in two variable logic over trees, 2013.
4. Mikolaj Bojanczyk, Anca Muscholl, Thomas Schwentick, and Luc Segoufin. Two-variable logic on data trees and xml reasoning. *J. ACM*, 56(3), 2009.
5. A. K. Chandra, D. Kozen, and L. J. Stockmeyer. Alternation. *J. ACM*, 28(1):114–133, 1981.
6. W. Charatonik and P. Witkowski. Two-variable logic with counting and trees. *Accepted for LICS*, 2013.
7. Kousha Etessami, Moshe Y. Vardi, and Thomas Wilke. First-order logic with two variables and unary temporal logic. *Inf. Comput.*, 179(2):279–295, 2002.
8. E. Grädel, P. Kolaitis, and M. Vardi. On the decision problem for two-variable first-order logic. *Bulletin of Symbolic Logic*, 3(1):53–69, 1997.
9. E. Grädel and M. Otto. On Logics with Two Variables. *Theoretical Computer Science*, 224:73–113, 1999.
10. E. Grädel, M. Otto, and E. Rosen. Undecidability results on two-variable logics. *Archiv für Mathematische Logik und Grundlagenforschung*, 38(4-5):313–354, 1999.
11. Katarzyna Idziak and Pawel M. Idziak. Decidability problem for finite heyting algebras. *J. Symb. Log.*, 53(3):729–735, 1988.
12. A.S. Kahr, E.F. Moore, and H. Wang. Entscheidungsproblem reduced to the $\forall\exists\forall$ case. *Proc. Nat. Acad. Sci. U.S.A.*, 48:365–377, 1962.
13. E. Kieroński. EXPSPACE-complete variant of guarded fragment with transitivity. In *STACS*, volume LNCS 2285, pages 608–619. Springer Verlag, 2002.
14. E. Kieroński. Decidability issues for two-variable logics with several linear orders. In *Computer Science Logic*, volume 12 of *LIPICs*, pages 337–351, 2011.
15. E. Kieroński and M. Otto. Small substructures and decidability issues for first-order logic with two variables. *Journal of Symbolic Logic*, 77:729–765, 2012.
16. Emanuel Kieronski, Jakub Michaliszyn, Ian Pratt-Hartmann, and Lidia Tendera. Two-variable first-order logic with equivalence closure. In *LICS*, pages 431–440. IEEE, 2012.
17. Maarten Marx. Xpath with conditional axis relations. In *EDBT*, volume 2992 of *Lecture Notes in Computer Science*, pages 477–494. Springer, 2004.
18. Maarten Marx. First order paths in ordered trees. In *ICDT*, volume 3363 of *Lecture Notes in Computer Science*, pages 114–128. Springer, 2005.
19. M. Mortimer. On languages with two variables. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, 21:135–140, 1975.

20. M. Otto. Two-variable first-order logic over ordered domains. *Journal of Symbolic Logic*, 66:685–702, 2001.
21. Michael O. Rabin. Decidability of second-order theories and automata on infinite trees. *Transactions of the American Mathematical Society*, 141:pp. 1–35, 1969.
22. Larry J. Stockmeyer. *The Complexity of Decision Problems in Automata Theory and Logic*. PhD thesis, MIT, Cambridge, Massachusetts, USA, 1974.
23. Wiesław Szostak and Lidia Tendera. The guarded fragment with transitive guards. *Ann. Pure Appl. Logic*, 128(1-3):227–276, 2004.
24. Philipp Weis. *Expressiveness and succinctness of first-order logic on finite words*. PhD thesis, University of Massachusetts Amherst, USA, 2011.

A Proof of Lemma 5

Proof. Assume that φ is satisfiable. By Lemma 3 and Lemma 4 there exists a small model $\mathfrak{T} \models \varphi$. The procedure accepts φ by making all its guesses in accordance to \mathfrak{T} , i.e. in the first step it sets $\bar{\alpha}$ to be equal to the full type of the root of \mathfrak{T} and then in each step it sets $\bar{\alpha}_i$ to be the full type of the i -th child of the previously considered element. In the opposite direction, from an accepting (tree-)run t of the procedure we can naturally construct a tree structure \mathfrak{T}_t , with 1-types of elements as guessed during the execution. Our procedure guesses actually not only 1-types but full types of elements. The function *locally-consistent* guarantees that the full types of elements in \mathfrak{T}_t are indeed as guessed. Since the procedure checks if each of those full types is φ -consistent, then by Proposition 1 we have that $\mathfrak{T}_t \models \varphi$. \square

B Proof of Lemma 6

Proof. Suppose $\mathfrak{T} \models \varphi[u]$, then there is $v \in \mathfrak{T}$ such that $\mathfrak{T} \models \psi[u, v]$. If $u \not\sim v$ then $\mathfrak{T} \models \psi_{\uparrow x \not\sim y}[u, v]$. By definition $u \in R'$ and thus there is $q \in Q$ such that $q \downarrow_* u$. The cases $v \downarrow_+ u$ and $u \downarrow_+ v$ are similar. If $u = v$ then $\mathfrak{T} \models \psi[u, u]$ and thus $\mathfrak{T} \models \psi_{\uparrow x=y}[u, u]$. In the opposite direction, suppose there is $q \in Q$ such that $q \downarrow_* u$. Notice that if $q \downarrow_* u$ then for every node v we have $q \not\sim v \Rightarrow u \not\sim v$. So if $q \in R'$ then there is a node v such that $\mathfrak{T} \models \psi_{\uparrow x \not\sim y}[u, v]$. Otherwise $q \in Q'$ and there exists a node v such that $\mathfrak{T} \models \psi_{\uparrow y \downarrow_+ x}[u, v]$. In both cases there is a node v such that $\mathfrak{T} \models \psi[u, v]$ and thus $\mathfrak{T} \models \varphi[u]$. The case if there is $p \in P$ such that $u \downarrow_* p$ is similar. If $\mathfrak{T} \models \psi_{\uparrow x=y}[u, u]$ then $\mathfrak{T} \models \psi[u, u]$ and thus $\mathfrak{T} \models \varphi[u]$. \square

C Proof of Lemma 7

Lemma 7 *Let $\varphi \in \text{FO}^2[\downarrow_+]$ over the alphabet τ_0 in ENNF and with one free variable, let \mathfrak{T} be a tree over the alphabet τ_0 , and let $a \in \tau_0$. There is a set $S \subseteq T$ which is a union of tree slices in \mathfrak{T} such that for every $i \in \mathfrak{T}^a : \mathfrak{T} \models \varphi[u]$ iff $u \in S$; and every path in \mathfrak{T} intersects at most $|\varphi|^2$ tree slices from S .*

Proof. Induction on the structure of φ . We consider only the case when $\varphi = \exists y \psi(x, y)$. Otherwise the proof is similar as in the corresponding Lemma 2.1.10 from [24]. Let

$$\psi(x, y) = \beta(x \downarrow_+ y, x = y, y \downarrow_+ x, x \not\downarrow y, \xi_1(x), \dots, \xi_s(x), \zeta_1(y), \dots, \zeta_t(y))$$

Applying the inductive hypothesis to the formulas ξ_σ , $\sigma \in [1, s]$, let S_σ be the set as described in the statement of this lemma, and let $I_{(\sigma, k_1)}, \dots, I_{(\sigma, k_\sigma)}$ be tree slices such that every path in \mathfrak{T} intersects at most $|\xi_\sigma|^2$ of them and $S_\sigma = \bigcup_{l=1}^{k_\sigma} I_{(\sigma, l)}$. We define the set $H = \bigcup_{\sigma=1}^s S_\sigma \cup \{r\} \cup L$, where r is the root of \mathfrak{T} and L is the set of leaves in \mathfrak{T} .

Looking at each tree slice I bounded by points from H , the truth values of the formulas ξ_1, \dots, ξ_s remain constant among all points from I^a . Let ξ_1^*, \dots, ξ_s^* be these respective true values. Thus, on all nodes from I^a , $\varphi(x)$ is equivalent to $\exists y \beta(x \downarrow_+ y, x = y, y \downarrow_+ x, x \not\downarrow y, \xi_1^*, \dots, \xi_s^*, \zeta_1(y), \dots, \zeta_t(y))$. This formula satisfies the requirements of Lemma 6, so that the truth of $\varphi(x)$ over I^a is determined by the relative position of x with respect to P , Q and by truth of the formulas $\zeta_1(x), \dots, \zeta_t(x)$ for the nodes in between P and Q . We now can construct the set S of all nodes from \mathfrak{T}^a where $\varphi(x)$ is true as the union of tree slices bounded by: points from H ; points that result from applying this lemma to the formulas $\zeta_1(x), \dots, \zeta_t(x)$; or points from P and Q added on every tree slice I .

We set a path in \mathfrak{T} and count the number of tree slices from S this path intersects. An intersection of a path from \mathfrak{T} with a tree slice is an interval. By the remark made after Lemma 6 we know there is at most one point from P and Q added on every path in I , thus there is at most one point $p \in P$ and $q \in Q$ on every interval. This means we can use the calculations in Lemma 2.1.10 from [24] to achieve at most $|\varphi|^2$ intervals on every path in \mathfrak{T} . \square

D Remaining part of the proof of Lemma 8

We argue that $\mathfrak{T}' \models \varphi$. To see this, we show by induction that for every sub-formula η of φ with at most one free variable and all $u \in T'$, $\mathfrak{T} \models \eta[u]$ iff $\mathfrak{T}' \models \eta[u]$.

If η is an atomic formula or a boolean combination of other formulas then the claim is obvious. Suppose $\eta = \varphi_\kappa$ for some $\kappa \in [1, k]$.

Suppose that $u \in I$ and $\mathfrak{T} \models \eta[u]$. Then there is a $v \in \mathfrak{T}$ such that $\mathfrak{T} \models \psi_\kappa[u, v]$. Let $I \in \mathcal{I}_\kappa^a$ such that $u \in I$. We find $\hat{v} \in T'$ such that $\mathfrak{T} \models \psi_\kappa[u, v]$ as follows: if $u \downarrow_+ v$ then there is a $\hat{v} \in P_I$ such that $v \downarrow_* \hat{v}$, if $v \downarrow_+ u$ then there is a $\hat{v} \in Q_I$ such that $\hat{v} \downarrow_* v$, if $u \not\downarrow v$ then there is a $\hat{v} \in R_I$ such that $v \downarrow_* \hat{v}$, if $u = v$ then we set $\hat{v} = u$. Clearly $\mathfrak{T}' \models \psi_\kappa[u, \hat{v}]$ and thus $\mathfrak{T}' \models \eta[u]$. Suppose now that $u \in T'$ and $\mathfrak{T}' \models \eta[u]$. Then it is easy to see that $\mathfrak{T} \models \eta[u]$.

So because $\mathfrak{T} \models \varphi$, we have $\mathfrak{T}' \models \varphi$. We now show that paths in \mathfrak{T}' have a bounded length. Set $I \in \mathcal{I}_\kappa^a$ and the formula $\psi_\kappa^I(x, y)$ as in fragment of this proof from the main part of the paper. For every $b \in \tau_0$ and every $i \in [1, t]$ let

$S_i'^b$ be a set as in Lemma 7 applied to the formula $\zeta_i(y)$, where $S_i'^b$ intersects at most $|\zeta_i|^2$ tree slices on every path. Thus there is a set \mathfrak{R}_κ^b of tree slices I' such that every path in \mathfrak{T} intersects at most $2 \cdot \sum_{i \in [1, t]} |\zeta_i|^2$ of them and $\bigcup \mathfrak{R}_\kappa^b = \mathfrak{T}$. Set a path in \mathfrak{T} and let \mathfrak{J}_κ^a be the set of intervals that are the intersections of this path with tree slices from \mathfrak{R}_κ^a . We claim that there is at most one element from P_I^b on every $J \in \mathfrak{J}_\kappa^b$. Suppose we have $u \in I$ and $v_1, v_2 \in J \cap P_I^b$. We show that $\mathfrak{T} \models \psi_{\kappa \downarrow x \downarrow_+ y}^I[u, v_1]$ iff $\mathfrak{T} \models \psi_{\kappa \downarrow x \downarrow_+ y}^I[u, v_2]$. Indeed recall that $\psi_{\kappa \downarrow x \downarrow_+ y}^I(x, y) = \beta(\top, \perp, \perp, \perp, \xi_1^I, \dots, \xi_s^I, \zeta_1(y), \dots, \zeta_t(y))$. Thus the boolean value of β depends on the boolean values of $\zeta_i(y)$. But we assumed that they are the same for v_1 and v_2 . Since P_I is a set of maximal nodes then $v_1 = v_2$. We can do analogous calculations for the sets Q_I and R_I . Altogether the length of every path in \mathfrak{T}' is at most $3 \cdot 2 \cdot \sum_{\kappa \in [1, k], a \in \tau_0, i \in [1, t]} |\zeta_i|^2 \leq 6 \cdot |\tau| \cdot |\varphi|^3$. \square

E Complexity of $\text{GF}^2[\downarrow_+]$ over singular trees

In this section we expand our arguments for PSPACE upper bound for $\text{GF}^2[\downarrow_+]$ over singular trees.

A $\text{GF}^2[\downarrow_+]$ formula φ is in *normal form* if $\varphi = \bigwedge_{i \in I} \forall xy (\eta_i(x, y) \Rightarrow \psi_i(x, y)) \wedge \bigwedge_{i \in J} \forall x (\lambda_i(x) \Rightarrow \exists y (\eta_i(x, y) \wedge \psi_i(x, y)))$, for some disjoint index sets I and J , where η_i is a guard of the form $x \downarrow_+ y$, $y \downarrow_+ x$ or $x = y$, $\lambda_i(x)$ is an atomic formula $a(x)$ for some unary symbol a , and $\psi_i(x, y)$ is a boolean combination of unary atomic formulas.

We can prove a slightly weaker counterpart of Lemma 1 for GF^2 . Namely, we show that satisfiability of a $\text{GF}^2[\downarrow_+]$ formula can be reduced to satisfiability of a normal form $\text{GF}^2[\downarrow_+]$ formula nondeterministically.

Lemma 9. *There exists a nondeterministic procedure $\text{GF}^2[\downarrow_+]$ -normalisation, such that for a $\text{GF}^2[\downarrow_+]$ formula φ over a signature τ , and a tree frame \mathcal{T} consisting of at least two nodes the following holds. The formula φ is satisfiable over \mathcal{T} (singularly satisfiable over \mathcal{T}) if and only if there exists a polynomial execution of $\text{GF}^2[\downarrow_+]$ -normalisation on φ producing a normal form $\text{GF}^2[\downarrow_+]$ formula φ' over a signature τ' consisting of τ and some additional unary symbols, satisfiable over \mathcal{T} (satisfiable over \mathcal{T} in a model which restricted to τ is singular).*

Proof. By the work from [23] it follows that for a given GF^2 formula φ over a signature τ there exists a polynomially computable formula $\varphi' = \bigwedge_{i \in I} ((\forall x r_i(x)) \Leftrightarrow \exists x (\lambda_i(x) \wedge \psi_i(x)) \wedge ((\forall x r_i(x)) \vee (\forall x \neg r_i(x)))) \wedge \bigwedge_{i \in J} \exists x (\lambda_i(x) \wedge \psi_i(x)) \wedge \varphi''$, for some disjoint index sets I and J , over a signature consisting of τ and some additional unary predicates, where $\lambda_i(x)$ is an atomic formula $a(x)$ for some unary symbol a , $\psi_i(x)$ is a boolean combinations of atoms, φ'' is in normal form, and none of r_i -s is used as a guard, such that φ and φ' are satisfiable over the same tree frames. Now for each $i \in I$ we guess whether $\forall x r_i(x)$ is satisfied or not and replace the occurrences of $r_i(x)$ and $r_i(y)$ in φ' by \top or \perp appropriately. We thus get a conjunction of a normal form formula, some formulas of the form $\exists x (\lambda_i(x) \wedge \psi_i(x))$, and some formulas of the form $\neg \exists x (\lambda_i(x) \wedge \psi_i(x))$. A formula

of the last type can be rewritten as $\forall xy(x = y \Rightarrow \neg\lambda_i(x) \vee \neg\psi_i(x))$. To deal with purely existential statements we introduce a fresh unary predicate *root* and make it true precisely at the root of a tree by adding the conjunct $\forall xy(x=y \Rightarrow (root(x) \Leftrightarrow \neg\exists y(y\downarrow_+x)))$. A formula $\exists x(\lambda_i(x) \wedge \psi_i(x))$ can be now rewritten as the normal form conjunct $\forall x(root(x) \Rightarrow \exists y(x\downarrow_+y \wedge (\lambda_i(y) \wedge \psi_i(y)) \vee (\lambda_i(x) \wedge \psi_i(x))))$. This transformation works properly over trees containing at least two nodes. The describe nondeterministic procedure is thus the required $\text{GF}^2[\downarrow_+]$ -**normalisation** procedure. \square

Let us see that in an arbitrary (not necessarily singular) model \mathfrak{T} of a normal form $\text{GF}^2[\downarrow_+]$ formula φ we can find a submodel in which the degree of nodes is bounded polynomially in $|\varphi|$ and in the length of the paths of \mathfrak{T} . As we are able to shorten paths in singular models to length polynomial in $|\varphi|$, this will lead to a polynomial bound on the degree of nodes in singular models of $\text{GF}^2[\downarrow_+]$ formulas (which, as we have seen, contrasts with the case of $\text{FO}^2[\downarrow_+]$).

Lemma 10. *Let φ be a normal form $\text{GF}^2[\downarrow_+]$ formula and let $\mathfrak{T} \models \varphi$. Then there exists a submodel $\mathfrak{T}' \models \varphi$ of \mathfrak{T} in which the number of successors of each node is bounded by $\max \cdot |\varphi|$, where \max is the length of the longest path in \mathfrak{T} .*

Proof. Let v be the root of \mathfrak{T} . For every conjunct φ_i of φ of the form $\forall x(\lambda_i(x) \Rightarrow \exists y(\eta_i(x, y) \wedge \psi_i(x, y)))$, with $\eta_i(x, y) = x\downarrow_+y$ pick a witness w for v and φ_i , mark w and mark all the elements u such that $\mathfrak{T} \models u\downarrow_+w$, i.e., the elements on the path from the root to w . Remove all subtrees rooted at successors of v containing no marked elements. Repeat this process for all the elements v of \mathfrak{T} , say, in the depth-first manner. Note that the structure obtained after each step is a model of φ , since we explicitly take care of providing lower witnesses, and the upper witnesses are retained automatically as every element which is not removed from the model is kept together with the whole path from the root from the original model \mathfrak{T} . Let \mathfrak{T}' be the structure obtained after the final step of the above procedure. Observe that the number of marked descendants of an element located at level l is bounded by $(l+1) \cdot |\varphi|$, thus the degree of each node of \mathfrak{T}' is bounded by $\max \cdot |\varphi|$ as required. \square

We recall the statement of Theorem 3 from the main part of the paper, and prove its part related to the upper bound. Lower bound is proved in the next section.

Theorem 3. *The satisfiability problem for $\text{GF}^2[\downarrow_+]$ over finite singular trees is PSPACE-complete.*

Proof. We show here that the problem belongs to PSPACE. We check satisfiability of a given $\text{GF}^2[\downarrow_+]$ formula φ over signature τ over singular trees in the following way. We run the non-deterministic procedure $\text{GF}^2[\downarrow_+]$ -**normalisation** (see Lemma 9) and obtain a formula φ' over signature τ' . On this formula we run a natural adaptation of the procedure $\text{FO}^2[\downarrow, \downarrow_+, \rightarrow, \rightarrow^+]$ -**sat-test** in which *maxdepth* is set to $6 \cdot |\tau| \cdot |\varphi|^3$, *maxdegree* is set to *maxdepth* $\cdot |\varphi'|$, the conditions verified in function *locally-consistent* are appropriately simplified (only $\bar{\alpha}(\theta_{\downarrow_+})$,

and $\bar{\alpha}(\theta_{\uparrow\uparrow+})$ are relevant this time), and which checks if all 1-type contain precisely one positive unary atom from τ .

The described alternating procedure works in time bounded polynomially in φ , so, as $\text{APTIME}=\text{PSPACE}$ [5], it can be also implemented to work in deterministic polynomial space. We claim that it accepts φ iff φ has a finite singular tree model. Assume that φ is accepted. This means that φ' has a tree model which restricted to τ is singular. By Lemma 9 it follows that φ has a singular model. In the opposite direction, let $\mathfrak{T} \models \varphi$ be singular, and let \mathcal{T} be the frame of \mathfrak{T} . By Lemma 8 we can assume that the depth of \mathfrak{T} is bounded by $6 \cdot |\tau| \cdot |\varphi|^3$. By Lemma 9, $\text{GF}^2[\downarrow_+]$ -normalisation can produce φ' which is satisfiable over \mathcal{T} , say in a model \mathfrak{T}' . By Lemma 4, φ' is also satisfied in a submodel \mathfrak{T}'' of \mathfrak{T}' in which the degree of every node is bounded by $6 \cdot |\tau| \cdot |\varphi|^3 \cdot |\varphi'|$. Thus our alternating procedure can make all its guesses in accordance to \mathfrak{T}'' and accept. \square

F Lower bounds for logics over singular trees

Theorem 5. *The satisfiability problem for $\text{FO}^2[\downarrow_+]$ over singular finite trees is NEXPTIME-hard .*

Proof. We give a reduction from the satisfiability problem of unary FO^2 , which is known to be NEXPTIME-complete (see e.g., [7]). For a given FO^2 formula φ over a unary signature τ we construct an equisatisfiable $\text{FO}^2[\downarrow_+]$ formula $T(\varphi)$ over the signature $\tau \cup \{\downarrow_+, elem\}$ where $elem$ is a fresh unary predicate. Without loss of generality we may assume that φ is built from variables x, y , unary predicate symbols, boolean connectives \wedge, \neg and existential quantification.

Now we inductively define the translation $T(\varphi)$.

$$\begin{aligned} T(p(x)) &= \exists y x \downarrow_+ y \wedge p(y) \\ T(\neg \varphi) &= \neg T(\varphi) \\ T(\varphi_1 \wedge \varphi_2) &= T(\varphi) \wedge T(\varphi_2) \\ T(\exists x \psi) &= \exists x elem(x) \wedge T(\psi) \end{aligned}$$

Note that $T(\varphi)$ is a formula of length linear in $(|\varphi|)$. It remains to be shown that φ and $T(\varphi)$ are equisatisfiable.

For one direction, assume that \mathfrak{A} is a model of φ . Construct a tree \mathfrak{T} such that all elements of the universe of \mathfrak{A} are immediate successors of the root of \mathfrak{T} and are labeled $elem$; each such element e has as many immediate successors as there are predicates in τ that are true of e , and each such successor is a leaf labeled with a distinct predicate true of e in \mathfrak{A} , see Figure 1. It can be easily proved by induction on the structure of φ that \mathfrak{T} is a (singular) model of $T(\varphi)$.

For the other direction assume that \mathfrak{T} is a model of $T(\varphi)$. Construct a structure \mathfrak{A} such that the universe of \mathfrak{A} is the set of nodes labeled $elem$ in \mathfrak{T} and for all elements e and all predicates p , $p(e)$ is true in \mathfrak{A} if and only if there is a node e' labeled p that is below e in \mathfrak{T} . Again it is easy to prove by structural induction that φ is true in \mathfrak{A} . \square

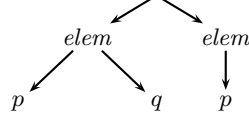


Fig. 1. Representation of a structure over the signature $\{p, q\}$. There are two elements in the universe; the first belongs to the relations p and q , the second to p .

Theorem 6. *The satisfiability problem for $\text{GF}^2[\downarrow_+, \downarrow]$ over singular finite trees is EXPSpace-hard.*

Proof. We give a reduction from $\text{GF}^2[\downarrow_+]$ over arbitrary trees. The idea of the encoding is the same as in Theorem 5: a node e in a tree is modeled by a singular node labeled $elem$ with immediate successors encoding predicates true in e . The binary predicate \downarrow_+ is used to preserve the structure of the tree, the additional \downarrow predicate gives the access to nodes modeling unary predicates. In the following reduction, for a given $\text{GF}^2[\downarrow_+]$ formula φ over a signature $\tau = \tau_0 \cup \{\downarrow_+\}$ we construct a $\text{GF}^2[\downarrow_+, \downarrow]$ formula over the signature $\tau \cup \{\downarrow, elem\}$ that is satisfiable over singular trees if and only if φ is satisfiable over trees.

Let us start with a formula ensuring that the underlying structure is an encoding of a tree. The formula *tree* is defined as the conjunction of

$$\bigwedge_{p \in \tau_0 \cup \{elem\}} \forall x p(x) \Rightarrow \forall y y \downarrow_+ x \Rightarrow elem(y)$$

with

$$\forall x elem(x) \Rightarrow \forall y x \downarrow_+ y \Rightarrow \bigvee_{p \in \tau_0 \cup \{elem\}} p(y).$$

It ensures that (unless the tree is trivial, i.e., no node is labeled at all) each node is labeled with some predicate symbol, all internal nodes are labeled $elem$ and only leaves may be labeled with predicates from τ_0 .

Without loss of generality we may assume that the formula φ is built from unary atoms, boolean connectives \wedge, \neg and guarded existential quantification. The translation $T(\varphi)$ of a formula φ is defined inductively as follows.

$$\begin{aligned} T(p(x)) &= \exists y x \downarrow y \wedge p(y) \\ T(\neg \varphi) &= \neg T(\varphi) \\ T(\varphi_1 \wedge \varphi_2) &= T(\varphi_1) \wedge T(\varphi_2) \\ T(\exists x p(x) \wedge \psi(x)) &= \exists x elem(x) \wedge T(p(x)) \wedge T(\psi(x)) \\ T(\exists y x \downarrow_+ y \wedge \psi(x, y)) &= \exists y x \downarrow_+ y \wedge elem(y) \wedge T(\psi(x, y)) \\ T(\exists y y \downarrow_+ x \wedge \psi(x, y)) &= \exists y y \downarrow_+ x \wedge elem(y) \wedge T(\psi(x, y)) \end{aligned}$$

Note that $T(\varphi)$ is a guarded formula of length linear in $(|\varphi|)$. Again a simple inductive argument shows that φ is satisfiable if and only if $tree \wedge T(\varphi)$ has a singular tree model. \square

Theorem 7. *The satisfiability problems for $GF^2[\downarrow_+, \rightarrow]$ and $GF^2[\downarrow_+, \rightarrow^+]$ over singular trees are EXPSpace-hard.*

Proof. We follow the construction from [13] and give a generic reduction from AEXPTIME. Consider an alternating Turing machine M working in exponential time. Without loss of generality we may assume that M works in time 2^n and that every non-final configuration of M has exactly two successor configurations. Let w be an input word of size n . Following [13] we construct a formula whose models encode accepting configuration trees of machine M on input w . In [13] each

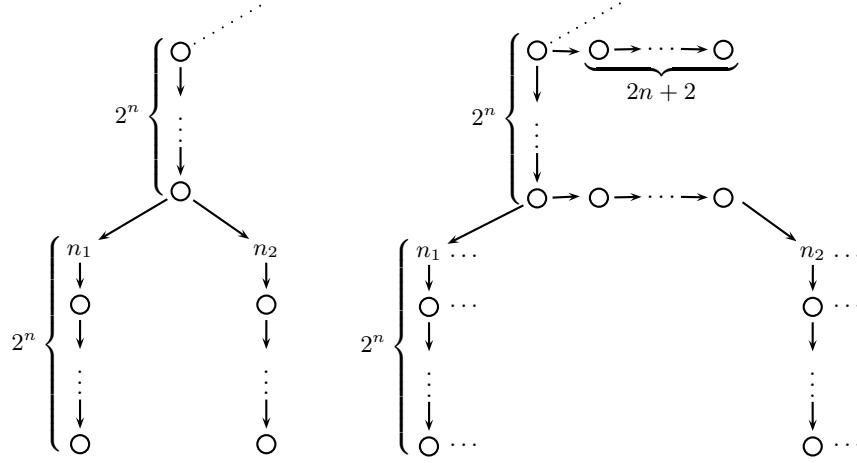


Fig. 2. Left: frame of a configuration tree in [13]; nodes n_1 and n_2 are siblings. Right: frame of a configuration tree in our encoding; nodes n_1 and n_2 are not siblings.

configuration is represented by 2^n elements of a tree, each of which represents a single cell of the tape of M (see left part of Figure 2). Each such node is then labeled with unary predicate symbols from the set $\{C_1, \dots, C_n, P_1, \dots, P_n\}$ to encode the number of a configuration (i.e., the depth of the configuration in the computation tree) and its position (i.e., the number of a cell) in the configuration: $C_i(x)$ is true if the i -th bit of the configuration number is 1 and $P_i(x)$ is true if the i -th bit of the position number is 1. Additional predicate symbols are used to encode the tape symbol and the state of the machine (if it is necessary, i.e., if the head of the machine is scanning the cell under consideration). Here, to encode the numbers, we use additional $2n$ elements that are siblings of the

node representing a cell, see right part of Figure 2. Each of these elements stores information about a single bit using one of two unary predicates *zero* or *one*. Then the atomic formulas $C_i(x)$ and $P_i(x)$ are simulated by formulas

$$\exists y x \rightarrow^+ y \wedge Path_i(y) \wedge one(y) \quad \text{and respectively} \quad \exists y x \rightarrow^+ y \wedge Path_{n+i}(y) \wedge one(y)$$

where the subformula $Path_i(y)$ is defined recursively as follows. For the logic $GF^2[\downarrow_+, \rightarrow]$ we define

$$\begin{aligned} Path_0(y) &= \neg \exists x x \rightarrow y \\ Path_{i+1}(y) &= \exists x x \rightarrow y \wedge Path_i(x) \end{aligned}$$

and for the logic $GF^2[\downarrow_+, \rightarrow^+]$ we define

$$\begin{aligned} Path_{\geq 0}(y) &= \neg \exists x x \rightarrow^+ y \\ Path_{\geq i+1}(y) &= \exists x x \rightarrow^+ y \wedge Path_{\geq i}(x) \\ Path_i(y) &= Path_{\geq i}(y) \wedge \neg Path_{\geq i+1}(y). \end{aligned}$$

Note that in both cases the formula $Path_i$ is guarded and has polynomial length. The negated atomic formulas $\neg C_i(x)$ and $\neg P_i(x)$ are simulated using predicate *zero* instead of *one*.

Now, having the ability to count, we may encode tape symbols and states of the machine by simply using more siblings, and we may follow the lines of the construction in [13] to encode the computation of M . The only remaining subtle point is that in [13] the two successor configurations are siblings in a computation tree while here they must not be siblings in order not to mess up the information about numbers — this may be simply done by rooting the two configurations at different nodes as shown on Figure 2. \square

Theorem 8. *The satisfiability problem for $GF^2[\downarrow_+]$ over singular trees is PSPACE-hard.*

Proof. We propose a reduction from the satisfiability of quantified boolean formulas, QBF. Let ψ be an instance of QBF problem. Without loss of generality we may assume that ψ is of the form

$$\exists v_k \dots \exists v_2 \forall v_1 \psi'$$

where the number of all quantifiers (k) is even, all even-numbered variables are existentially quantified, all odd-numbered variables are universally quantified and ψ' is a propositional formula over the variables v_1, \dots, v_k .

We now translate the formula ψ to a formula over the signature

$$\tau = \{root, leaf, true, false, \downarrow_+\}$$

such that ψ is true if and only if its translation is satisfiable over singular trees.

First, for $i \in \{0, \dots, k\}$ we define auxiliary formulas $depth_i$ and $height_i$. Let $depth_0(x) = root(x)$ and for $i \geq 1$ let $depth_i(x) = \exists y x \downarrow_+ y \wedge depth_{i-1}(y)$. Intuitively, the formula $depth_i(x)$ expresses that the node x occurs at distance at least i from the root. Let $height_0(x) = leaf(x)$, $height_1(x) = depth_k(x)$ and let $height_i(x) = depth_{k+1-i}(x) \wedge \neg depth_{k+2-i}(x)$ for $i > 1$. For $i > 0$ the formula $height_i(x)$ expresses that x is a node at depth exactly $k+1-i$; in the construction below, for $i \geq 0$, the formula $height_i(x)$ will mean that the subtree rooted at x has height i . Note that $height_i(x)$ is a guarded formula of length linear in i .

In the following construction a model of the translation of ψ is a tree that describes a set of valuations justifying that ψ is true. It is a binary tree of depth $k+1$ where every path describes a valuation of variables v_1, \dots, v_k . Every node at height i is labeled either *true* or *false*, which corresponds to a value of the variable v_i under a given valuation. Every non-leaf node at odd height i has two successors corresponding to the universally quantified variable v_{i+1} ; every node at even height i where $i > 0$ has one successor corresponding to the existentially quantified variable v_{i+1} . If $k > 0$ then let $tree_k$ be the conjunction of

$$\exists x root(x), \quad (1)$$

$$\forall x root(x) \Rightarrow (\exists y x \downarrow_+ y \wedge height_k(y) \wedge (true(y) \vee false(y))), \quad (2)$$

$$\begin{aligned} \forall x true(x) \Rightarrow (& height_i(x) \Rightarrow \\ & ((\exists y x \downarrow_+ y \wedge height_{i-1}(y) \wedge true(y)) \\ & \wedge (\exists y x \downarrow_+ y \wedge height_{i-1}(y) \wedge false(y)))) \\ & \text{for all even numbers } 2 \leq i \leq k, \end{aligned} \quad (3)$$

$$\begin{aligned} \forall x false(x) \Rightarrow (& height_i(x) \Rightarrow \\ & ((\exists y x \downarrow_+ y \wedge height_{i-1}(y) \wedge true(y)) \\ & \wedge (\exists y x \downarrow_+ y \wedge height_{i-1}(y) \wedge false(y)))) \\ & \text{for all even numbers } 2 \leq i \leq k, \end{aligned} \quad (4)$$

$$\begin{aligned} \forall x true(x) \Rightarrow (& height_i(x) \Rightarrow \\ & \exists y x \downarrow_+ y \wedge height_{i-1}(y) \wedge (true(y) \vee false(y))) \\ & \text{for all odd numbers } 3 \leq i < k, \end{aligned} \quad (5)$$

$$\begin{aligned} \forall x false(x) \Rightarrow (& height_i(x) \Rightarrow \\ & \exists y x \downarrow_+ y \wedge height_{i-1}(y) \wedge (true(y) \vee false(y))) \\ & \text{for all odd numbers } 3 \leq i < k, \end{aligned} \quad (6)$$

$$\begin{aligned} \forall x true(x) \Rightarrow (& height_1(x) \Rightarrow \\ & (\exists y x \downarrow_+ y \wedge leaf(y))), \end{aligned} \quad (7)$$

$$\begin{aligned} \forall x false(x) \Rightarrow (& height_1(x) \Rightarrow \\ & (\exists y x \downarrow_+ y \wedge leaf(y))). \end{aligned} \quad (8)$$

In the case of $k = 0$ the formula $tree_0$ boils down to $\exists x root(x) \wedge \forall x root(x) \Rightarrow (\exists y x \downarrow_+ y \wedge leaf(y))$. Note that $tree_k$ is a guarded formula of length polynomial in k . Now we inductively define the translation $T(\psi')$ of the quantifier-free

formula ψ' .

$$T(true) = true$$

$$T(false) = false$$

$$T(v_i) = \exists y \ y \downarrow_+ x \wedge height_i(y) \wedge true(y)$$

$$T(\neg\varphi) = \neg T(\varphi)$$

$$T(\varphi_1 \wedge \varphi_2) = T(\varphi) \wedge T(\varphi_2)$$

$$T(\varphi_1 \vee \varphi_2) = T(\varphi) \vee T(\varphi_2)$$

Note that $T(\psi')$ is a guarded formula of length polynomial in $(|\psi'| + k)$. It is not difficult to prove by induction on k (and by nested structural induction on propositional formulas with free variables v_1, \dots, v_k) that ψ is true if and only if $tree_k \wedge \forall x \ leaf(x) \Rightarrow T(\psi')$ has a singular tree model. Each node labeled *leaf* in such a model uniquely determines a path to a node labeled *root* and such a path corresponds to a valuation of the variables v_1, \dots, v_k that makes the formula ψ' true. \square